MANEJO BÁSICO DE CONTENEDORES

ps	Ver todos los contenedores que se estén ejecutando.
ps -a	Ver todos los contenedores existentes.
pull <imagen>	Descargar una imagen.
run <banderas> <imagen> <comando> <atributos>	Formato de ejecución de contenedores.
run -ti <imagen> <comando>	Ejecución en modo interactivo.
attach <contenedor>	Entrar a un contenedor que se esté ejecutando.
run --name <nombre> <imagen>	Asignar nombre a un contenedor.
start <banderas> <contenedor>	Iniciar un contenedor apagado.
stop <contenedor>	Detener un contenedor.
kill <contenedor>	Forzar la detección de un contenedor.
rm <contenedor>	Eliminar un contenedor que esté apagado.
run -d <imagen>	Ejecutar contenedor en segundo plano.
logs <contenedor>	Ver los registros que deja el contenedor.
exec <banderas> <contenedor> <comando>	Ejecutar un comando en un contenedor en ejecución.
run -P <imagen>	Mapeamiento de puertos automaticamente.
pause <contenedor>	Pausar un contenedor.
unpause <contenedor>	Continuar la ejecución de un contenedor pausado.
inspect <contenedor>	Ver información detallada de un contenedor.
rmi <imagen>	Eliminar una imagen.
stats <contenedor>	Ver la cantidad de recursos consumido del contenedor.
system prune	Borra todos las dependencias docker no usadas.
run -ti -v <ruta_host>:<ruta_imagen> <imagen>	Crear contenedor con volúmenes.
run -p <puerto_host>:<puerto_contenedor>	Enlazar puerto del host con puerto de un contenedor.

CREACIÓN DE IMÁGENES (Dockerfile)

docker build <ruta>	Generar una imagen a partir de un Dockerfile.
docker history <imagen>	Ver todas las capas que componen una imagen.
COPY <archivos> <ruta_imagen>	Copiar archivos dentro de la imagen.
ADD <url> <ruta_imagen>	Descargar o copiar archivo dentro de la imagen.
EXPOSE <puerto>	Abrir(exponer) un puerto en la imagen.
WORKDIR <ruta_imagen>	Indicar una ruta relativa por defecto.
MAINTAINER <nombre>	Referir el nombre de quien mantiene la imagen.
CMD [<>, <>, <>, ...]	Ejecutar un comando por defecto al crear la imagen.
FROM <imagen>	Imagen base para la creación de la imagen nueva.

REDES

network ls	Ver listado de redes de docker.
run --net=host <imagen>	Asignar la IP del host al contenedor.
network create <nombre>	Crear una nueva red.
network connect <red> <contenedor>	Añadir un contendor a más redes.

OPTIMIZAR CONTENEDORES

Extraer el contenido de un contenedor y los guarda en otro. En el proceso se borran todas las capa de la imagen de dicho contenedor y lo optimiza a una única capa. *export <imagen> | import <imagen>*.

CONECTAR CONTENEDORES EN RED

1. Un contenedor por defecto se crea mediante la red tipo *bridge*.
2. Los contenedores deben tener un nombre predefinido.
3. Al crear alguno de los contenedores, uno de ellos deberá pasarle mediante la bandera *link* el nombre del contenedor a enlazar en red. *run --link <contenedor> <imagen>*.
4. Los contenedores pueden hacerse *ping* mediante el nombre del contenedor. Este comportamiento solo se presenta en las redes tipo *bridge*.

SEGURIDAD (TLS Cliente-Demonio)

SERVIDOR

1. Crear llaves públicas y privadas de la entidad certificante.
    a. Crear carpeta docker-ca donde se almacenarán todas las llaves y certificados.
    b. *openssl genrsa -aes256 -out ca-key.pem 2048*.
    c. Ingresar contraseña de las llaves.
2. Crear el certificado.
    a. *openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem*.
    b. Ingresar la contraseña de las llaves públicas y privadas.
    c. Pedirá datos de metadata. Pueden ser ignorados.
3. Crear llaves públicas y privadas del servidor.
    a. *openssl genrsa -out server-key.pem 2048*.
    b. Ingresar contraseña de las llaves.
4. Crear requerimiento de firma CSR para el servidor.
    a. *openssl req -subj "/CN=localhost" -new -key server-key.pem -out server.csr.*
    b. En caso de estar creando los certificados en un servidor remoto, se debe cambiar el CN de localhost al nombre de dominio del servidor.
5. Crear archivo indicador de la dirección IP del servidor.
    a. echo "subjectAltName = IP:127.0.0.1" > extfile.cnf.
    b. En caso de estar creando los certificados en un servidor remoto, se debe cambiar la dirección IP 127.0.0.1 a la correspondiente del servidor.
6. Firmar la clave del servidor.
    a. openssl x509 -req -days 365 -in server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out server-cert.pem extfile extfile.cnf.

CLIENTE

1. Crear las llaves públicas y privadas del cliente.
    a. *openssl genrsa -out client-key.pem 2048*.
    b. Ingresar contraseña de las llaves.
2. Crear requerimiento de firma CSR para el cliente.
    a. *openssl req -subj '/CN=client' -new -key client-key.pem -out client.csr*.
3. Crear archivo de configuración del cliente.
    a. *echo "extendedKeyUsage = clientAuth" > extfile.cnf*.
4. Firmar la clave del cliente.
    a. *openssl x509 -req -days 365 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out client-cert.pem extfile extfile.cnf*.